

# 4 RS  
PATENT  
5702D-6844  
09/13/98  
08/07/98

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

TANAKA, Yoshiyuki; YATABE, Makoto;  
SATO, Takeaki; and KAWAMOTO, Kazuya  
Serial No: unassigned  
Filed: 07 August 1998  
For: METHOD FOR CONTROLLING NON-  
VOLATILE SEMICONDUCTOR MEMORY  
SYSTEM

) Art Unit: unassigned  
) Examiner: unassigned  
)  
)  
)  
)  
)  
)  
)  
)  
)

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to:  
Assistant Commissioner for Patents  
Washington D.C. 20231, on

August 7, 1998  
Date of Deposit

Louis A. Mok  
Name

Signature

8/7/98  
Date

TRANSMITTAL OF PRIORITY DOCUMENT

Assistant Commissioner for Patents  
Washington, D.C. 20231

Dear Sir:

Enclosed herewith are certified copies of Japanese patent application No. 9-214561 filed August 8, 1997 and 10-119099 filed April 28, 1998, from which priority is claimed under 35 U.S.C. §119 and Rule 55.

Acknowledgment of the priority documents is respectfully requested to ensure that the subject information appears on the printed patent.

Respectfully submitted,

LOEB & LOEB LLP

By:

Louis A. Mok

Registration No. 22,585  
Attorney for Applicant(s)

Date: August 7, 1998

10100 Santa Monica Blvd., 22nd Floor  
Los Angeles, California 90067-4164  
Telephone: (310) 282-2000  
Facsimile: (310) 282-2192

日 本 国 特 許 庁

PATENT OFFICE  
JAPANESE GOVERNMENT

JCS40 U.S. PRO  
09/130818



別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
in this Office.

願 年 月 日  
Date of Application:

1997年 8月 8日

願 番 号  
Application Number:

平成 9年特許願第214561号

願 人  
Applicant(s):

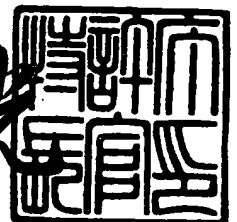
株式会社東芝

CERTIFIED COPY OF  
PRIORITY DOCUMENT

1998年 3月 6日

特許庁長官  
Commissioner,  
Patent Office

荒井寿光



出証番号 出証特平10-3011085

【書類名】 特許願

【整理番号】 46A9750371

【提出日】 平成 9年 8月 8日

【あて先】 特許庁長官殿

【国際特許分類】 G11C 7/00

【発明の名称】 不揮発性半導体メモリの制御方法

【請求項の数】 7

【発明者】

【住所又は居所】 神奈川県川崎市幸区堀川町580番1号 株式会社東芝  
半導体システム技術センター内

【氏名】 田中 義幸

【発明者】

【住所又は居所】 神奈川県川崎市幸区堀川町580番1号 株式会社東芝  
半導体システム技術センター内

【氏名】 谷田部 誠

【発明者】

【住所又は居所】 神奈川県川崎市幸区堀川町580番1号 株式会社東芝  
半導体システム技術センター内

【氏名】 佐藤 雄亮

【発明者】

【住所又は居所】 神奈川県川崎市幸区堀川町580番1号 株式会社東芝  
半導体システム技術センター内

【氏名】 河本 和也

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100083161

【弁理士】

【氏名又は名称】 外川 英明

【電話番号】 03-3457-2512

【先の出願に基づく優先権主張】

【出願番号】 平成 8年特許願第226913号

【出願日】 平成 8年 8月28日

【手数料の表示】

【予納台帳番号】 010261

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9605503

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 不揮発性半導体メモリの制御方法

【特許請求の範囲】

【請求項1】 システムが管理する論理ブロックと、

複数のメモリセルにより構成され、前記論理ブロックに対応するデータを記憶する物理ブロックと、

前記物理ブロック中に含まれ、前記対応する論理ブロックのアドレスを記憶する冗長部と、

前記物理ブロックが少なくとも2以上で構成される物理ブロックエリアとを具備し、前記論理ブロックと前記物理ブロックエリアとの対応関係を管理するための論理番地／物理番地変換テーブルを作成することを特徴とするメモリシステムの制御方法。

【請求項2】 メモリアクセス時に、前記論理番地／物理番地変換テーブルを参照し前記論理ブロックに対応する物理ブロックエリアのアドレスを読み出し、前記物理ブロックエリアを構成する少なくとも2以上の物理ブロックの前記冗長部に記憶される対応する論理ブロックのアドレスを読み出すことにより、前記論理ブロックと対応する物理ブロックを選択することを特徴とする請求項1記載のメモリシステムの制御方法。

【請求項3】 電源投入時に、前記論理番地／物理番地変換テーブルを作成することを特徴とする請求1記載のメモリシステムの制御方法。

【請求項4】 第1の所定単位によって管理されるファイルを、第二の所定単位に分割された記憶領域において記憶する半導体記憶装置において、

前記第1の所定単位の境界が、前記第2の所定単位の境界上に配置されるよう制御することを特徴とするメモリシステムの制御方法。

【請求項5】 前記第2の所定単位は消去単位であることを特徴とする請求項4記載のメモリシステムの制御方法。

【請求項6】 システムが管理するファイルと、

前記ファイルの内容を記憶するデータ領域と、

前記ファイルとデータ領域の対応関係を記憶する管理領域とを有し、

前記ファイルの消去を行う時に、前記管理領域に対応するデータ領域が空き領域である旨をマークし、対応する前記データ領域の消去を行うことを特徴とするメモリシステムの制御方法。

【請求項7】 システムが管理するファイルと、

前記ファイルの内容を記憶するデータ領域と、

前記ファイルとデータ領域の対応関係を記憶する管理領域とを有し、

前記ファイルの消去を行う時に、前記管理領域に対応するデータ領域が空き領域である旨をマークしておき、メモリシステムに入力される信号に基づいて、前記管理領域の内容を検知し対応するデータ領域の消去を行うことを特徴とするメモリシステムの制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、不揮発性半導体メモリの制御方法に関し、特に不揮発性半導体メモリカードの制御に使用されるものである。

【0002】

【従来の技術】

近年図11に示すようなフラッシュメモリカードがデジタルスチールカメラやPDA等の携帯情報機器の記憶媒体として注目されている。このメモリカードは薄型のプラスチックパッケージ1にわずかな窪みが設けられておりその窪みに22ピンの平面電極を有するフラッシュメモリ2が埋め込まれている。本フラッシュメモリカードは専用のコネクタを介してホストシステムに電氣的に接続され、データの入出力を行う。

【0003】

フラッシュメモリとしてNAND型フラッシュメモリと呼ばれるフラッシュメモリが使用され販売がなされている。

16MビットのNAND型フラッシュメモリの場合を例にとるとフラッシュメモリは512個の物理的なメモリブロックに分割されている。このブロックは消去時の最小単位となっている。1ブロックはさらに16ページに分割される。1

ページは書き込みおよび読み出しの基本的な単位となる。1 ページは264 バイトから構成され、うち256 バイトはユーザーデータ領域（データ部）、残りの8 バイト（冗長部）はエラー訂正符号および管理情報等の格納（冗長部）に使用される。

## 【0004】

通常パソコン等ではデータはセクタ（512 バイト）単位で管理されるため、本メモ리카ードでも512 バイト単位でのデータ管理を基本とすべく2 ページをペアとして取り扱う。

## 【0005】

このようなメモ리카ードの制御では、データ更新時は消去済み領域に更新データを書き込み、元のデータが存在する領域を消去するという、追加書き込み方式を採用しているため、ある論理ブロックに対応するデータが存在する物理ブロックは、固定ではなく、常にメモリ内を移動している。したがって、上述のごとく物理ブロックの冗長部には自分がどの論理ブロックに対応するデータを保持しているかを示す論理ブロックアドレス情報を記憶している。

## 【0006】

ゆえに、通常は電源投入時に、全物理ブロックの該論理ブロックアドレス情報格納領域をサーチし、システムRAM上に、論理ブロックと物理ブロックの変換テーブルを作る。一度テーブルを作成した後は、該テーブルを参照すれば、論理ブロックに対応する物理ブロックがすぐに判断可能なため、全ブロックのサーチ動作は電源投入時に1回行われる。当然のことながら、データの更新を行い、対応する物理ブロックの位置が変化した場合、メモリシステムは、テーブルの更新作業を行い、次のアクセスに備えることになる。

## 【0007】

従来のテーブルを図12に示す。ここでは、論理ブロックは500個設定されており、1個の論理ブロックは連続した8セクタに相当する。すなわち論理ブロック0は論理セクタ0～7を意味する。この連続した8セクタ分のデータは、メモリ中の512ブロックのうちのいずれかの物理ブロックに存在する。512個の物理ブロックから1個のブロックを選択するには、9ビット必要となる。ソフ

トウェアの利便性に配慮してオフセットがそのまま物理ブロックを指し示す様にテーブルを構成すると、1ブロックに対し2バイト、計1KBのRAM領域が必要となる。例えば、論理ブロック5の情報が格納されている物理ブロックの番地はテーブルの先頭からオフセット5ワード（10バイト）目に格納される。

## 【0008】

このように、従来用いられていた方法においては、テーブルが必要とするRAM領域が非常に大きいという問題点があった。通常よく用いられる、汎用CPUでは内蔵RAMとして1KB程度のRAMを搭載しているのが一般的である。従って、従来は、テーブルだけで1KBを使用する必要があり、内蔵RAMのみではシステム構成ができず、外づけのRAMをシステムとして有することが条件となりコストアップの大きな要因となっていた。

## 【0009】

次に、フラッシュメモリカードをパソコン等で使用する場合の制御方法を64MビットNAND型フラッシュメモリを例に説明する。フラッシュメモリは1000個の物理的なメモリブロックに分割されている。このブロックは消去時の最小単位となっている。1ブロックはさらに16ページに分割される。1ページは書き込みおよび読み出しの基本的な単位となる。1ページは528バイトから構成され、うち512バイトはユーザーデータ領域（データ部）、残りの16バイト（冗長部）はエラー訂正符号および管理情報等の格納（冗長部）に使用される。この場合のテーブルの例を図13に示す。論理ブロックは1000個に設定されており、1個の論理ブロックは連続した16セクタに相当する。すなわち論理ブロック0は論理セクタ0～15を意味する。

## 【0010】

以下に、フラッシュメモリの書き込み、消去について簡単に述べる。フラッシュメモリの書き込みはページと呼ばれる単位で一括で実行される。64MビットNAND型EEPROMの場合は1ページ528バイトである。また消去はブロック単位で実行される。64MビットNAND型EEPROMは16ページで1ブロックを構成する。このように64MビットNAND EEPROMでは書き込みと消去の単位が異なる。従ってあるページのみを消去してデータを更新する



事は出来ない。

【0011】

フラッシュメモリカードをパソコンで使用する場合には一般的にDOS支配下のドライブとして取り扱われる。図14に従来のDOSフォーマットパラメータを示す。ここでは、図14(a)として、クラスタサイズ4KBの場合、図14(b)としてクラスタサイズが8KBの場合を示している。クラスタとはDOSのファイル管理の基本最小単位で、ファイルサイズが非常に小さくても1個のクラスタ分の容量は占有する。ファイルサイズが大きい場合は、複数のクラスタのチェーンとして管理され、その管理情報はFAT(File Allocation Table)に格納される。クラスタのサイズやFATの管理方法等はブートセクタと呼ばれるセクタ内で管理される。1個のデバイスが複数のドライブとして管理される場合は、マスターブートセクタにその情報が格納される。ファイルの書き込みとしては、OSからクラスタ単位で書き込み命令が発行される。

【0012】

図14(a)に示すように、クラスタサイズが4KBの場合、論理セクタの0にマスターブートセクタ、論理セクタ16にブートセクタ、論理セクター17～22にFAT、論理セクター23～28にFATのコピー、論理セクタ29～44にディレクトリー、論理セクター45以降にファイルデータ領域が配置されている。

【0013】

図14(b)に示すように、クラスタサイズが8KBの場合、論理セクタの0にマスターブートセクタ、論理セクタ16にブートセクタ、論理セクター17～19にFAT、論理セクター20～22にFATのコピー、論理セクタ23～38にディレクトリー、論理セクター39以降にファイルデータ領域が配置されている。

【0014】

まず、図15を用いて、クラスタサイズ4KBの場合を例に従来の書き換えシーケンスを説明する。クラスタサイズは4KBなので、連続した8セクタ分の書き込み命令がOSから発行される。このとき論理セクタの45～52(クラスタ

A) に対する書き込み（データ更新）が発生し、論理セクタの77～85（クラスタA'）に書き換わる。

- 1) 消去済み新領域を探し、論理セクター32～44までを元ブロックから新領域にコピーする。
- 2) 論理セクター45～47の新データを新領域に書き込む。
- 3) 元ブロックを消去する。
- 4) アドレス変換テーブルの更新。
- 5) 消去済み新領域を探し、論理セクター48～52の新データを新領域に書き込む。
- 6) 元ブロックの論理セクター53～63のデータを新領域にコピーする。
- 7) 元ブロックを消去する。
- 8) アドレス変換テーブルの更新。

従って、外部からみて8セクタの書き換えを実施した場合、実際デバイスとしては、合計32セクタ（32ページ）の書き込み動作と、2ブロックに対する消去動作が実施されたことになる。

#### 【0015】

次に、図16を用いて、クラスタBに対する書き込みシーケンスを説明する。この場合論理セクタの53～60（クラスタB）に対する書き込み（データ更新）が発生し、論理セクタの69～76（クラスタB'）に書き換わる。

- 1) 消去済み新領域を探し、論理セクター48～52までを元ブロックから新領域にコピーする。
- 2) 論理セクター53～60の新データを新領域に書き込む。
- 3) 論理セクター61～63までを元ブロックから新領域にコピーする。
- 4) 元ブロックを消去する。
- 5) アドレス変換テーブルの更新。

従って、外部からみて8セクタの書き換えを実施した場合、実際デバイスとしては、合計16セクタ（16ページ）の書き込み動作と、1ブロックに対する消去動作が実施されたことになる。

#### 【0016】

次に、図17を用いて、クラスタサイズ8KBの場合を例に従来の書き込みシーケンスを説明する。クラスタサイズは8KBなので、連続した16セクタ分の書き込み命令がOSから発行される。このとき論理セクタの39～54（クラスタA）に対する書き込み（データ更新）が発生し、論理セクタの71～86（クラスタA'）に書き換わる。

- 1) 消去済み新領域を探し、論理セクター32～38までを元ブロックから新領域にコピーする。
- 2) 論理セクター39～47の新データを新領域に書き込む。
- 3) 元ブロックを消去する。
- 4) アドレス変換テーブルの更新。
- 5) 消去済み新領域を探し、論理セクター48～54の新データを新領域に書き込む。
- 6) 元ブロックの論理セクター55～63のデータを新領域にコピーする。
- 7) 元ブロックを消去する。
- 8) アドレス変換テーブルの更新。

従って、外部からみて16セクタの書き換えを実施した場合、実際デバイスとしては、合計32セクタ（32ページ）の書き込み動作と、2ブロックに対する消去動作が実施されたことになる。

#### 【0017】

上記クラスタが4KBのものと8KBの場合を同じ8KBのデータが書き込まれる場合を想定し比較すると、クラスタサイズが4KBの場合は、2回の書き込みに処理が分割され合計48セクタの書き込みおよび3ブロック分の消去動作が発生する。これに対し、クラスタサイズが8KBの場合は、1回の書き込みに処理が集約され合計32セクタの書き込みおよび2ブロック分の消去動作が発生することになる。

#### 【0018】

このように従来は、外部から見て更新されたセクタ数に比較し、実際にデバイス上で実行された書き込み動作および消去動作がはるかに多くなるため、外部からみた書き換え速

度が遅くなると言う問題点があった。

【0019】

さらに、通常DOSのファイルシステムにおいては、ファイル消去コマンド実行時に、ディレクトリー上に該当ファイルが無効である旨のマーキングを行い、該当ファイルが占有していたメモリ領域をFAT(File Allocation Table)上で開放する。従って、ファイル本体のデータ部分はフラッシュメモリ上に消去されずに残っている。消去コマンド実行時の管理領域とデータ領域の関係を図22に示す。図22において、例えばFile-1とFile-4の消去コマンドを実行した場合、管理領域のFile-1とFile-4が開放されdel. markがマーキングされる。このときデータ領域においてはFile-1、File-4は消去されない。

【0020】

このため、その後の書込みコマンド実行時に、前記開放された領域に対し新たなファイルのデータ部分を書き込まれる際に、まずフラッシュメモリの消去動作が必要となる。このため、ファイル書き込み時に必ずフラッシュメモリの消去動作を伴い、ファイル書き込み速度を劣化させるという問題点もあった。

【0021】

【発明が解決しようとする課題】

本発明は上記問題を鑑みてなされたものであり、論理番地と物理番地の変換テーブルで必要となるRAMの容量を低減し、汎用CPUの内蔵RAMのみでフラッシュメモリを制御する方法を提供し、これにより従来必要であった外付けRAMを不要となし、大幅なコストダウンを実現することを目的とする。

【0022】

さらに、物理ブロックにおいて、DOS上のファイル管理の基本単位であるクラスタの区切れが、消去の単位となるブロックをまたがないようにする方法を提供し、データの高速書込みを実現することを目的とする。

【0023】

また、消去コマンド実行時に物理ブロックの管理領域を開放すると同時に、そのデータ領域の消去も行ふことにより、その後の書込みコマンド実行時の処理速

度の向上を実現することを目的とする。

【0024】

【課題を解決するための手段】

上記目的を達成する手段として、本発明では、システムが管理する論理ブロックと、複数のメモリセルにより構成され、前記論理ブロックに対応するデータを記憶する物理ブロックと、前記物理ブロック中に含まれ、前記対応する論理ブロックのアドレスを記憶する冗長部と、前記物理ブロックが少なくとも2以上で構成される物理ブロックエリアとを具備し、前記論理ブロックと前記物理ブロックエリアとの対応関係を管理するための論理番地／物理番地変換テーブルを作成するメモリシステムの制御方法を提供する。

【0025】

また、上記メモリシステムの制御方法において、メモリアクセス時に、前記論理番地／物理番地変換テーブルを参照し前記論理ブロックに対応する物理ブロックエリアのアドレスを読み出し、前記物理ブロックエリアを構成する少なくとも2以上の物理ブロックの前記冗長部に記憶される対応する論理ブロックのアドレスを読み出すことにより、前記論理ブロックと対応する物理ブロックを選択するメモリシステムの制御方法を提供する。

【0026】

さらに、前記前記論理番地／物理番地変換テーブルを電源投入時に、作成するメモリシステムの制御方法を提供する。

また、データの高速書き換えを実現する手段として、本発明ではシステムが管理するファイルと、少なくとも1以上の前記ファイルを記憶する複数のブロックとを有し、前記ブロック内で、前記ファイルの先頭が前記ブロックの先頭部と一致するよう配置されることを特徴とするメモリシステムの制御方法を提供する。

【0027】

さらに、前記ブロックは消去時の最小単位であることを特徴とするメモリシステムの制御方法を提供する。

また、消去エリアへのデータの高速書込みを実現する手段として、本発明では

システムが管理するファイルと、前記ファイルの内容を記憶するデータ領域と、前記ファイルとデータ領域の対応関係を記憶する管理領域とを有し、前記ファイルの消去を行う時に、前記管理領域に対応するデータ領域が空き領域である旨をマークし、対応する前記データ領域の消去を行うことを特徴とするメモリシステムの制御方法を提供する。また、消去エリアへのデータの高速書込みを実現する第2の手段として、本発明ではシステムが管理するファイルと、前記ファイルの内容を記憶するデータ領域と、前記ファイルとデータ領域の対応関係を記憶する管理領域とを有し、前記ファイルの消去を行う時に、前記管理領域に対応するデータ領域が空き領域である旨をマークしておき、メモリシステムに入力される信号に基づいて、前記管理領域の内容を検知し対応するデータ領域の消去を行うことを特徴とするメモリシステムの制御方法を提供する。

【0028】

## 【発明の実施の形態】

図1に本フラッシュメモリカードを使用するための制御に関するメインフローを示し、以降各行程の動作について述べる。

(1-A) 挿入検知について以下に述べる。

【0029】

メモリカードに対する処理をスタートするためには、正常にコネクタに挿入されていることが前提条件となる。不完全な挿入状態でメモリカードにアクセスを行うと物理的またはデータの破壊を引き起こす可能性があるからである。従って、本発明のメモリシステムの制御フローでは、メモリカードが正常に挿入されているかを検出する手段を持っている。その手段として、機構的なもの、電気的なもの等がある。

【0030】

次に(1-B)電源電圧検知について以下に述べる。

本メモリカードには、5V電源電圧と3.3V電源電圧の製品があるが、3.3V電源電圧動作のメモリカードに5Vの電源が印加されると、電圧破壊等の問題が発生する可能性がある。これを回避するため、システムは電源電圧を検知している。検知方法として外観で判断する方法と、電気的に判断する方法が考えら

れる。

【0031】

第3に（1-C）容量検知について述べる。

フラッシュメモ리카ードには記憶容量またはインタフェース仕様の異なる複数種類のものがある。システムにメモ리카ードが挿入された時には、デバイスのメーカーコードおよびデバイスコード等を判別し、想定外のコードの場合は新たなアクセスはしないようにする。また、メーカーコード、デバイスコード等の読み出しには正規の電源電圧を投入する。

【0032】

第4に（1-D）物理フォーマット確認の確認について述べる。

メモ리카ードでは、データを記憶するための物理フォーマットがなされており、メモ리카ードがシステムに挿入された時はその物理フォーマットを確認し、もしサポート外のフォーマットがなされていた場合には、データを破壊することなくリジェクトする。また、未知の物理フォーマット品が挿入された場合に、システムが再度物理フォーマットを実行することは、フラッシュメモリの先天性、および後天性の不良ブロックに対する処理が不完全になる危険性があるため、注意が必要である。例えば16MビットのNAND型フラッシュメモリを例に考えると、フラッシュメモリは512個の物理的なブロックに分割されており、先頭ブロックには物理フォーマットの形式やカードの属性情報等が書き込まれている。残りのブロックはデータ領域として使用されるので、先頭ブロックのデータを判別する事によって、サポート可能な物理フォーマットがなされているかどうかを判断しても良い。

【0033】

第5に（1-E）論理フォーマットはメモ리카ードをDOS上のデバイスとしてアクセス可能とするために行われる。DOSのクラスタの切れ目とNANDフラッシュメモリの物理的なブロックの区切れを一致させると、より高速な動作が可能となる。

【0034】

第6の（2-F）各動作は、システムのメモリアクセス動作を示し、読み出し

、書込み、消去動作のことである。

第7の(2-G) 抜去検知は前述した挿入検知同様、抜去の検知も行うものである。抜去が検知された場合、メモリシステムはその動作の終了する。

#### 【0035】

次に、本発明のメモリシステムの、物理ブロックの構造と論理ブロックの関係について説明する。

図2に16MビットNAND型フラッシュメモリの場合の物理構造を示す。メモリは512個のブロックに分割されている。このブロックは消去時の最小単位となっている。1ブロックは16ページから構成される。1ページは書き込み、読み出しの基本的な単位となる。1ページは264バイトから構成され、うち256バイトはユーザーデータ領域(データ部)、残りの8バイトはエラー訂正符号および管理情報等の格納(冗長部)に使用される。

#### 【0036】

通常パソコン等ではデータはセクタ(512バイト)単位で管理されるため、本メモリカードでも512バイト単位でのデータ管理を基本とし、2ページをペアとする。データ領域の内部データ構成を図3に示す。未使用の正常ブロックは、データ部、冗長部とも“FFh”に設定されている。下記に各々のバイトの意味あいにつき説明する。Data Area-1は512バイトデータのうち、前半の0~255バイトのデータが格納される。Data Area-2には512バイトデータのうち、後半の256~511バイトのデータが格納される。User Data Areaのデータは、ユーザに解放されており使用方法はユーザに一任される。Data Status Areaはデータが正常か否かを示す。通常は“FFh”だが、正常でないデータが書き込まれている場合に“00h”が設定される。Block Status Areaはブロックが良か不良かを示す。通常は“FFh”だが、不良ブロックの場合、“00h”(初期不良ブロック)、“F0h”(後発不良ブロック)が設定される。2ビット以上“0”があった場合は、不良ブロックであると判断する。なお、本データは同一ブロック内では全て同じ値を書き込む。はブロックの論理アドレス情報を示す。なお、1論理ブロックを構成する8セクタには512の物理ブロックのうち1物



理ブロックが相当するので、本データは同一ブロック内では全て同じ値が書込まれることになる。同様に、Block Address Area-2はBlock Address Area-1のデータと同じ内容が書かれている。ECC Area-1は偶数ページデータ(256バイト)の3バイトECCコードである。ECC Area-2は、奇数ページデータ(256バイト)の3バイトECCコードである。

## 【0037】

ここで、ECCについて実例をもとに説明する。本実施例では256バイト(2048ビット)単位のデータに対しECC符号を生成する。1ビット訂正の機能を持たせるために、256バイトに対して22ビットのECCデータを使用している。256バイトは図4に示すように、データに並べたものである。

## 【0038】

即ち、1Byte目の入力のbit0が2048bitの1bit目(アドレス;00000000 00)となり、256Byte目の入力のbit7が2048bitの2048bit目(アドレス;11111111 111)となる。

## 【0039】

ECCコード(ラインパリティ(LP)とカラムパリティ(CP))は図5に示す条件を満足する1024ビットの奇数パリティとして算出される。

カラムパリティCP0~CP5は1バイト(8ビット)のデータが入力される毎に更新される。ECC符号の生成をソフトウェアで行う場合は、1バイトの入力(256通り)に対してのカラムパリティ計算結果を予めシステム内のROM上に持つ方法が考えられる。こうすることにより、ビット単位の演算が不要になり、計算時間が大幅に短縮できる。またバイトの入力(256通り)に対してのカラムパリティ計算結果を電源投入時に一括で計算し、RAM上に保持する方法も考えられる。前案に比較しROMは無くても済むが、代わりにRAM領域が必要となる。

## 【0040】

次に、実際のデータ転送に関して説明する。前述したように、Block A

address Area-1はブロックの論理アドレス情報を示している。16 MビットのNAND型フラッシュメモリの場合、物理ブロックは連続した8論理セクタ分のデータを有する。例えば、論理セクタ0～7のデータは、メモリ中のいずれかの物理ブロック内にそのデータを保持している。論理セクタと物理番地を対応させるためのデータとして上述のBlock Address Areaがある。論理セクタ0～7のデータが格納された物理ブロックの該当エリアには、自分が論理セクタ0～7の情報を記憶している旨の論理アドレス情報が格納されている。

#### 【0041】

論理セクタ0～7のデータが保持されている物理ブロックの番地は固定ではない。例えば、ホストから論理セクタ0～7のデータの更新命令が来たときには、フラッシュメモリ上の消去済みのエリアに更新データを書き込み、元のデータの有ったブロックは消去する。従って、論理セクタ0～7のデータを格納していた物理番地は変更されることになる。すなわち、論理セクタ0～7のデータはメモリのどの領域に格納されているかは、各物理ブロックのBlock Address Areaを参照して検索をする必要がある。アクセスのたびに検索作業を行うことは、処理速度の面から考えて妥当ではない。したがって、本発明では電源投入時に全ブロックをサーチして、論理番地と物理番地との変換テーブルを作ることになる。一度テーブルを作成した後は、該テーブルを参照すれば、論理ブロックに対応する物理ブロックがすぐに判断可能なため、全ブロックのサーチ動作は電源投入時1回で良い。当然のことながら、データの更新を行い、対応する物理ブロックの位置が変化した場合は、テーブルの更新作業を行い、次のアクセスに備える。

#### 【0042】

次に、上述したメモ리카ードの制御フローにおいて、(1-F)のメモ리카ードの各動作について図面を参照に説明する。

図6に電源投入時の制御フローチャートを示す。以下にフローに従い説明を加える。

(6-1) コネクタにメモ리카ードが挿入されたことを受けて、電源電圧の検

知を行う。

(6-2) メモリカードのIDコードを読み出し、記憶容量を読みとる

(6-3) もしシステムがサポートしていないIDコードが読み出されたり  
ジェクトする。

(6-4) 物理フォーマットの確認を行う。物理番地の先頭ブロックの情報を  
読む。

(6-5) もしシステムがサポートしていないフォーマットがなされていれば  
リジェクトする。

(6-6) 論理番地と物理番地の変換テーブルを作るとともに、次の書き込み  
動作で書き込みを行うための消去済み領域を選択する。

#### 【0043】

図7に上記(6-6)に示した電源投入時に作成する論理番地と物理番地の変換テーブルの作成のフローを示す。このフローにしたがって作成された論理番地／物理番地変換テーブルを図8に示す。図8に示す物理ブロックエリア(Physical Block Area)とは、連続した2個の物理ブロックの集合体を意味する。例えば、物理ブロックエリア0とは、物理ブロック0と物理ブロック1を示す。本テーブルでは1個の論理ブロックに対して1個の物理ブロックエリアを割り付けている。例えば、論理ブロック0に対し、物理ブロックエリア5が割り振られている場合、物理ブロックの10もしくは11が実際の論理ブロック0のデータを記憶していることになる。従って、実際にアクセスする時には、物理ブロック10および11の冗長部の論理アドレスとの関連を示すデータ領域を検索し、どちらが論理ブロック0のデータを本当に格納しているのかを判断する必要がある。しかし、極めて限定された領域をリードするのみで済むため、メモリアクセスの性能に及ぼす影響はほとんどない。

#### 【0044】

このとき、物理ブロックエリアは全体で256個(512/2)存在し、8ビットのデータによって記述する事が可能となる。ソフトウェアの利便性に配慮してオフセットがそのまま物理ブロックを指し示す様にテーブルを構成すると、1ブロックに対して1バイト、計0.5KBのRAM領域が必要となる。例えば、

論理ブロック5の情報が格納されている物理ブロックエリアの番地はテーブルの先頭からオフセットバイト目に格納される。

【0045】

この0.5KBのRAM容量は、従来必要であった1KBのRAM容量に対し半分となる。汎用CPUは通常1KB程度のRAM領域をもっているが、本実施例によって得られた0.5KBのRAM領域の削減は非常に大きな割合をしめる。

【0046】

すなわちコストアップを招く外付けRAMを設けなくても、本実施例によって得られた0.5KBの空き領域を使用する事によってシステムを構成する事が可能となりコストダウンをはかることができる。

【0047】

また、上記実施形態に限ず、物理ブロックエリアとして4物理ブロックを定義しても良いし、さらに大きなブロック数を想定しても構わない。

以下に、図7に示した電源投入時に作成する論理番地／物理番地の変換テーブルの作成のフローについて説明する。

(7-1) 論理ブロック／物理ブロック変換テーブルが入るRAM領域をリセットする。

(7-2) 次のデータ書き込みに使用される消去済みのブロックを記憶するテーブル領域をリセットする。

(7-2) 物理ブロックの1からサーチを開始する。

(7-3) ブロックの冗長部を読み出す。

(7-4) 所定領域のデータを元に該ブロックが正常なブロックか否かを判断する。不良ブロックの場合以下に続く処理は不要となり、次のブロックのサーチに移行する。

(7-5) 消去済みのエリアかどうかを判断する。

(7-6) 消去済みのブロックであれば次の書き込みの際に使用するブロックの候補としてテーブル上に格納する。

(7-8) 消去済みのエリアで無ければ、論理アドレス情報領域を抽出する。

このとき、パリティチェックを行い妥当性を確認する。

(7-9) 上記内容をもとに論理アドレスと物理アドレスの変換テーブルを作成する。

(7-10) 物理ブロック番号のカウントアップ。

(7-11) 512ブロックサーチしたら終了。

【0048】

次に、読み出し時の動作フローを図9を参照に説明する。

(9-1) 読み出しを行う先頭セクタアドレスと転送セクタ数をホストから受け取る。

(9-2) 読み出し範囲が、妥当な範囲か検証する。

(9-3) セクタを論理ブロックに変換。16Mビット品の場合は1ブロック8セクタ構成なので8で割ることになる。

(9-4) 論理アドレス／物理アドレス変換テーブルを参照し、該当論理ブロックが存在する物理ブロックエリアを得る。

(9-5) ブロックエリアにある2個の物理ブロックの論理アドレス情報領域を調べ、いずれがホストの指定した論理ブロックのデータを格納しているかを調べる。

(9-5) 特定された物理ブロックから1セクタ分データを読み出す。例えばセクター番号が0の時は、物理ブロックの先頭2ページのデータを読み、例えばセクター番号が7の時は、物理ブロックの最終2ページのデータを読む。1個の物理ブロックの中では、8個のセクタのデータが順番に並んでいる。

(9-8) 読み出したデータに対しエラーチェックを行い、エラーが無いか確認する。

(9-8) エラーが検出されたら、訂正可能か否かを判断する。

(9-9) エラーが検出され、なおかつ訂正可能な場合データを訂正する。

(9-10) ホストが要求したセクタ数を読み出したら終了する。

(9-11) 次の読み出しセクターが物理ブロックの境界を越えるかどうかを判断する。例えば、セクター7から8に移行する場合、データは各々異なる物理ブロックに存在するため、新たに論理アドレス／物理アドレス変換テーブルを再

度参照する。

(9-12) 同一ブロック内で読み出しを継続する場合は、読み出すページのカウンタアップする。

(9-3) 別のブロックの移動する場合は、論理ブロックをカウンタアップし、ページのカウンタもリセットする。

【0049】

次に、書込み時の動作について説明する。

書き込みは基本的に次の3部分の処理に大別される。例えば、論理セクタ3を書き換える場合例に説明する。更新はセクタ3のみだが、セクタ0から7の8セクタ分のデータは同一ブロック上に存在するので、1ブロックに対する処理が必要になる。

【0050】

第1に、論理セクタ0、1及び2に関しては、データの更新は無ので、論理セクタ0、1及び2のデータは、元々格納されていた物理ブロックから新しく書き込みを行う物理ブロックへコピーされる。

【0051】

第2に、論理セクタ3は更新するので、元のデータをコピーする必要はなく、ホストから与えられたデータを新たに書き込みを行うブロックへ書き込む。

第3に、論理セクタ4～7はデータの更新はないので、論理セクタ4～7のデータは、元々格納されていた物理ブロックから新しく書き込みを行う物理ブロックへコピーする。

以上のように、1ブロックに対して、コピー／更新データ書き込み／コピーの動作が基本となる。勿論、書き込みがセクタ0～7の様な場合は、1ブロック分全てのデータが更新されるのでコピー動作は不要となることは自明である。以下に示すフローチャートの分岐は主にこれから書き込むセクタが更新データか、もしくはコピー動作なのかを判断しながら進行することになる。

【0052】

次に、書込み時の動作について図10を参照に説明する。

(10-1) ホストからデータの更新を行う先頭セクタ番地と転送セクタ数を

受け取る。

(10-2) 論理ブロック番号に変換し、論理番地と物理番地の変換テーブルを参照する。読み出し動作と同様に物理ブロックエリアの2個のブロックから本当の物理ブロックを選択する。ここで選択したブロックからコピーすべきデータを吸い上げることになる。

(10-3) 物理ブロックの先頭から処理を開始する。

(10-4) ブロック前半のコピー動作かもしくはデータの更新かを判断する。

(10-5) コピーであれば、元ブロックからデータを読み出し、新ブロックに書き込みを実行する。

(10-6) 次のセクタの処理に移行する。

(10-7) (10-4) で更新領域と判断されたら、ホストから受け取った更新データを基に書き込みを行う。

(10-8) 次のセクタの処理に移行する。

(10-9) ホストが要求するセクタ数書き込んだか確認する。

(10-10) (10-9) で要求数の書き込みが終了したと判断された場合、ブロックの境界か否かを判断する。未書き込みの領域が残っていれば、ブロック後半のコピー動作に移行。ブロック境界であれば、これ以上コピー動作を実行する必要はない。

(10-11) 元のブロックからデータを読み出し、新ブロックに書き込む。

(10-12) 次のセクタの処理に移行する。

(10-13) (10-4) でホストの要求するセクタ数の書き込みが終了していない場合は更なる書き込みが必要だが、ブロック境界であれば、次の物理ブロックに対する処理に移行する。

(10-14) 次にブロック処理に移行または、処理を終了する前に、書き込みを行った結果をもとに論理アドレス/物理アドレス変換テーブルの更新を行い、さらに元データが存在した物理ブロックを消去し、次の処理での新たな書き込み領域としての候補領域として登録する。

(10-15) 次のブロックの処理に移行する。

【0053】

次に、本願発明のフラッシュメモリカードをDOSフォーマット形式で用いた場合に、データの書き換え動作を高速化するための実施例を説明する。

図18に本発明によるDOSフォーマットパラメータを示す。クラスタサイズが4KBおよび8KBの場合とも、クラスタの区切れが物理的なブロックの区切れをまたがないようパラメータが設定されていることにある。これは、DOSフォーマットパラメータの内、ブートセクタの配置される場所を調整することにより実現できる。

【0054】

まず、図19を用いて、クラスタサイズ4KBの場合を例に書き込みシーケンスを説明する。

クラスタサイズは4KBなので、連続した8セクタ分の書き込み命令がOSから発行される。このとき論理セクタの48～55（クラスタA）に対する書き込み（データ更新）が発生し、論理セクタの64～71（クラスタA'）に書き換わる。

- 1) 消去済み新領域を探し、論理セクタ48～55までの新データを新領域に書き込む
- 2) 論理セクタ56～63の元データを新領域にコピーする。
- 3) 元ブロックを消去する。
- 4) アドレス変換テーブルの更新。

【0055】

従って、外部からみて8セクタの書き換えを実施した場合、実際デバイスとしては、合計16セクタ（16ページ）の書き込み動作と、1ブロックに対する消去動作が実施されたことになる。

【0056】

次に、図20を用いて、またクラスタBに対する書き込みシーケンスを説明する。

この場合論理セクタの56～63（クラスタB）に対する書き込み（データ更新



）が発生し、論理セクタの72～79（クラスタB'）に書き換わる。

- 1) 消去済み新領域を探し、論理セクタ48～55までの元データを新領域にコピーする。
- 2) 論理セクタ56～63の新データを新領域に書き込む。
- 3) 元ブロックを消去する。
- 4) アドレス変換テーブルの更新。

従って、外部からみて8セクタの書き換えを実施した場合、実際デバイスとしては、合計16セクタ（16ページ）の書き込み動作と、1ブロックに対する消去動作が実施されたことになる。

#### 【0057】

次に、図21を用いて、クラスタサイズ8KBの場合を例に書き込みシーケンスを説明する。クラスタサイズは8KBなので、連続した16セクタ分の書き込み命令がOSから発行される。このとき論理セクタの48～63（クラスタA）に対する書き込み（データ更新）が発生し、論理セクタの64～79（クラスタA'）に書き換わる。

- 1) 消去済み新領域を探し、論理セクタ48～63の新データを新領域に書き込む。
- 2) 元ブロックを消去する。
- 3) アドレス変換テーブルの更新。

従って、外部からみて16セクタの書き換えを実施した場合、実際デバイスとしては、合計16セクタ（16ページ）の書き込み動作と、1ブロックに対する消去動作が実施されたことになる。

#### 【0058】

上記クラスタが4KBのものと8KBの場合を同じ8KBのデータが書き込まれる場合を想定し比較すると、クラスタサイズが4KBの場合は、2回の書き込みに処理が分割され合計32セクタの書き込みおよび2ブロック分の消去動作が発生する。これに対し、クラスタサイズが8KBの場合は、1回の書き込みに処理が集約され合計16セクタの書き込みおよび1ブロック分の消去動作が発生することになる。

## 【0059】

従来のメモリシステムの書き換え速度と比較すると、クラスタが4KBの場合では、8KBのデータを更新するのに48セクタの書き込みおよび3ブロック分の消去動作が発生していたものが、32セクタの書き込みおよび2ブロック分の消去動作となり、時間的に2/3に短縮されたことになる。また、クラスタが8KBの場合では、8KBのデータを更新するのに32セクタの書き込みおよび2ブロック分の消去動作が発生していたものが、16セクタの書き込みおよび1ブロック分の消去動作となり、時間的に1/2に短縮されたことになる。

## 【0060】

このように、DOSのファイル管理の単位であるクラスタの区切れが、フラッシュメモリの物理的なブロックの境界をまたがないことにより、書き換え速度の高速化がはかれる。

## 【0061】

また、クラスタのサイズに着目するとクラスタが4KBの場合では、8KBのデータを更新するのに32セクタの書き込みおよび2ブロック分の消去動作が発生していたものが、クラスタが8KBの場合では16セクタの書き込みおよび1ブロック分の消去動作となり、時間的に1/2になる。すなわち、クラスタのサイズをフラッシュメモリの物理ブロックのサイズと同じにすることによってより高速な書き込みが可能となる。勿論、クラスタのサイズがフラッシュメモリの物理ブロックのサイズの整数倍の場合も同様の効果が得られることが可能である。

## 【0062】

次に、本願発明のフラッシュメモリカードの書き込み性能を向上させる実施例を説明する。

本願発明のフラッシュメモリカードシステムは、通常DOSのファイルシステムにおけるファイル消去と異なり、ディレクトリー上に該当ファイルが無効である旨のマーキングを行い、該当ファイルが占有していたメモリ領域をFAT(File Allocation Table)上で開放するのみならず、ファイル本体のデータ部分をフラッシュメモリ上で消去することを特徴とする。すなわちファイルの消去命令時

に、開放されたクラスタの領域に対する消去動作を実施する。消去コマンド実行時の管理領域とデータ領域の関係を図23に示す。図23において、例えばFile-1とFile-4の消去コマンドを実行した場合、管理領域のFile-1とFile-4が開放されdel. markがマーキングされ、さらにデータ領域においてFile-1とFile-4が記憶されていた領域は消去される。

#### 【0063】

従って、次に新たなファイル書き込み命令が発生したときに選択されるクラスタは既に消去済みであるので、即座に書き込みが可能となり、ファイル書き込み速度が向上する。一般にフラッシュメモリにおいては書き込みより消去の方が時間がかかるため、本実施例によって達成されるファイル書き込み速度の向上効果は顕著である。

#### 【0064】

また本実施例における効果が最も顕著に現れる条件は実施例1から分かる様に、クラスタサイズとフラッシュメモリのブロックサイズを一致させた場合である。クラスタサイズがフラッシュメモリのブロックサイズより小さい場合は、ブロックの一部分を消去することになるが、処理が煩雑であることに加え、仕様上ブロックの一部分のみが消去されている状態を許さない場合もある。クラスタサイズとブロックサイズが一致していれば、単純にブロックを消去する事によってクラスタを開放できる。勿論クラスタサイズが物理ブロックサイズの整数倍の場合も同様の効果が得られる。

#### 【0065】

また、本発明はその主旨を変えない範囲で様々拡張が可能である。例えば、上記実施例では、ファイル消去時にデータ領域の該当するクラスタへの消去動作を実行したが、消去動作の実行タイミングはそれに限られない。例えば、フォーマット動作を実行する時点で全クラスタの消去動作を実行しても良い。勿論メモ리카ードの出荷時点でデータ領域のクラスタを消去済みの状態で出荷すると良い。通常メモ리카ードの出荷時にはメモ리카ードの出荷試験を実施する。このテスト終了時にディレクトリーおよびFATの書き換えによって、ファイルが無い状態

にするだけでなく、本実施例のようにデータ領域の消去動作を実行しておく、エンドユーザーの手に渡った時点で、ユーザーが手を加えることなく高速なファイル書き込みが期待できる。さらに例えばデジタルスチルカメラ等で使用する場合を考えると、カメラ上でのファイルの消去、再フォーマット等の画像ファイルの消去を伴う動作時において、FAT等の書換のみならず、ファイル本体が格納されていたクラスタ領域の消去を同時に実行すれば、その後の画像書き込みシーケンスにおいて、高速書き込みが可能となり、連写や動画の取り込み等が可能となる。この際には、クラスタサイズがデバイスのブロックサイズの正数倍で、クラスタの区切れがブロックサイズの区切れと一致しているとファイル本体部分の消去が容易となる。またデジタルカメラ等で電源の投入時等に自動的に不要なファイル部分を消去するようにしても良い。こうすれば、パソコン上で単純にFAT等の更新によってファイル消去がなされていたメモ리카ードに対しても、ユーザーの手も煩わせることなく書き込み高速化が計れる。またそのタイミングは電源投入時にのみならずいつでもかまわない。また、本発明はコンピュータシステム上で本メモ리카ードを使用する場合も同様の手段により書き込みの高速化が計れる。

#### 【0066】

また、最後に本発明によれば、データの書換時に発生する、消去および書き込み回数を減らすことが可能なため、一般的に書換回数に制限のあるフラッシュメモリを使用する場合には、メモリの寿命をのばす効果が書換の高速化と共に含まれていることを述べておく。

#### 【0067】

##### 【発明の効果】

以上、説明して来たように、本発明のメモリシステムの制御方法を用いることにより、RAM領域の大幅な削減が実現できる。これは、従来のメモ리카ードシステムは論理アドレス／物理アドレス変換テーブルで必要なRAM容量が大きく、汎用CPUの内蔵RAMのみではシステム構成ができず、外づけのRAMを設けることが条件となっていたに対して、本発明のメモ리카ードシステムは論理番

地と物理番地の変換テーブルとして必要なRAMの容量を低減し、汎用CPUの内蔵RAMのみ制御可能としたフラッシュメモリを制御する方法を用いるからである。このことにより、従来必要であった外付けRAMを不要とし、大幅なコストダウンを実現することを可能となる。

【0068】

また、本発明のフラッシュメモリカードは、DOSのファイル管理の単位であるクラスタの区切れが、フラッシュメモリの物理的なブロックの境界をまたがないことにより、データの書換時に発生する消去および書き込み回数を減らすことが可能なため、書き換え速度の高速化が可能となる。さらに、一般的に書換回数に制限のあるフラッシュメモリを使用する場合には、メモリの寿命をのばすことも可能となる。

【図面の簡単な説明】

【図1】

本発明のメモリシステムの制御方法を用いたメモリシステムのメインフローチャートを示す図である。

【図2】

本発明のメモリシステムにおける物理ブロック構成を示す図である。

【図3】

本発明のメモリシステムにおけるページ構成を示す図である。

【図4】

本発明のメモリシステムにおけるECCデータ構成を示す図である。

【図5】

本発明のメモリシステムにおけるECCコードを算出するための条件を示した図である。

【図6】

本発明のメモリシステムにおける電源投入時の制御フローチャートである。

【図7】

本発明のメモリシステムにおける論理アドレス／物理アドレス変換テーブル作成のフローチャートである。

【図 8】

本発明のメモリシステムにおける論理アドレスと物理アドレスの変換テーブル構成を示す図である。

【図 9】

本発明のメモリシステムにおける読み出し時のフローチャートである。

【図 10】

発明のメモリシステムにおける書込み時のフローチャートである。

【図 11】

メモリカード外観図である。

【図 12】

従来のメモリシステムにおける論理アドレスと物理アドレスの変換テーブル構成を示す図である。

【図 13】

64MビットNAND型フラッシュメモリシステムにおける論理アドレスと物理アドレスの変換テーブル構成を示す図である。

【図 14】

従来のDOSフォーマットのパラメータを示す図である。

【図 15】

従来の書き換えシーケンスを示す図である。

【図 16】

従来の書き換えシーケンスを示す図である。

【図 17】

従来の書き換えシーケンスを示す図である。

【図 18】

本発明のDOSフォーマットのパラメータを示す図である。

【図 19】

本発明の書き換えシーケンスを示す図である。

【図 20】

本発明の書き換えシーケンスを示す図である。

【図21】

本発明の書き換えシーケンスを示す図である。

【図22】

従来の消去コマンド実行時の管理領域とデータ領域の関係を示した図である。

【図23】

本発明の消去コマンド実行時の管理領域とデータ領域の関係を示した図である。

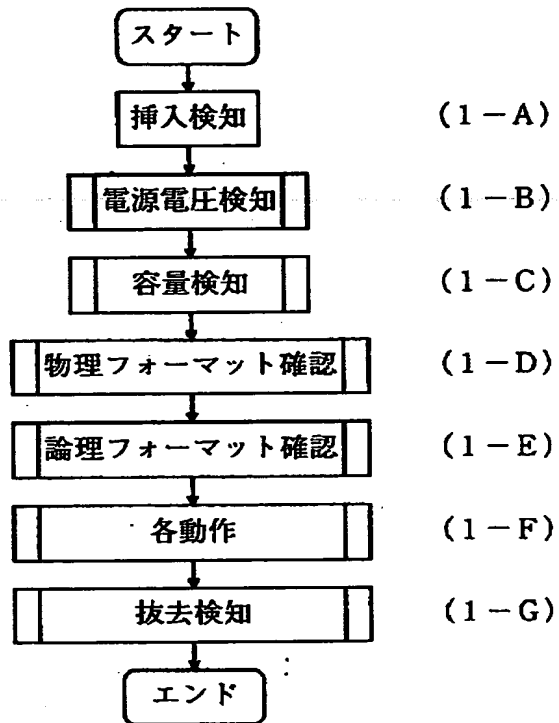
。

【符号の説明】

- 1           メモ리카ードのプラスチックパッケージ
- 2           フラッシュメモリ

【書類名】 図面

【図 1】





【図 2】

		0	255	256	263
Block 0	Page 0	データ領域 (256バイト)		冗長部 (8バイト)	
	Page 1				
	⋮				
	Page 15				
Block 1	Page 0				
	Page 1				
	⋮				
	Page 15				
⋮	⋮	⋮		⋮	
⋮	⋮	⋮		⋮	
⋮	⋮	⋮		⋮	
⋮	⋮	⋮		⋮	
Block 511	Page 0				
	Page 1				
	⋮				
	Page 15				

【図 3】

◎データ部

バイト	0ページ (偶数ページ)	1ページ (奇数ページ)
0		
～	DATA Area-1	DATA Area-2
255		

◎冗長部

バイト	偶数ページ	奇数ページ
256	User Data Area	ECC Area-2
257		
258		Block Address Area-2
259		
260	Data Status Area	ECC Area-1
261	Block Status Area	
262	Block Address Area-1	
263		

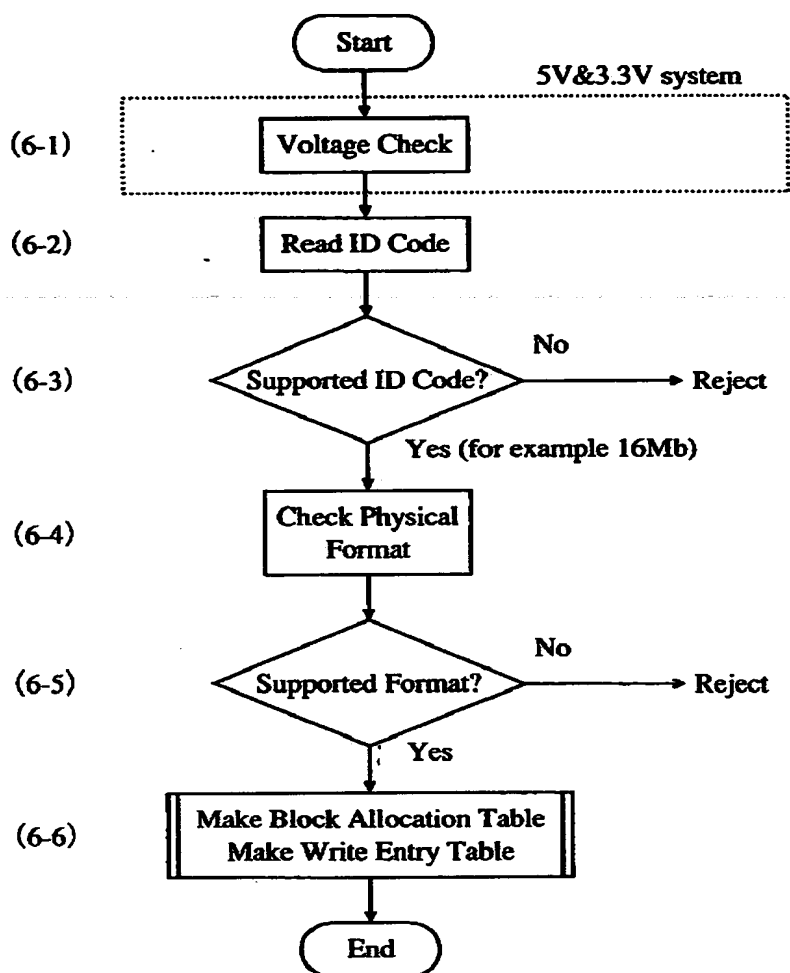
【図 4】

bit 7	bit 6		bit 1		bit 0	
1stByte	00000000	111	00000000	110	00000000	001
2ndByte	00000000	111	00000001	110	00000001	001
...						
255thByte	11111110	111	11111110	110	11111110	001
266thByte	11111111	111	11111111	110	11111111	001

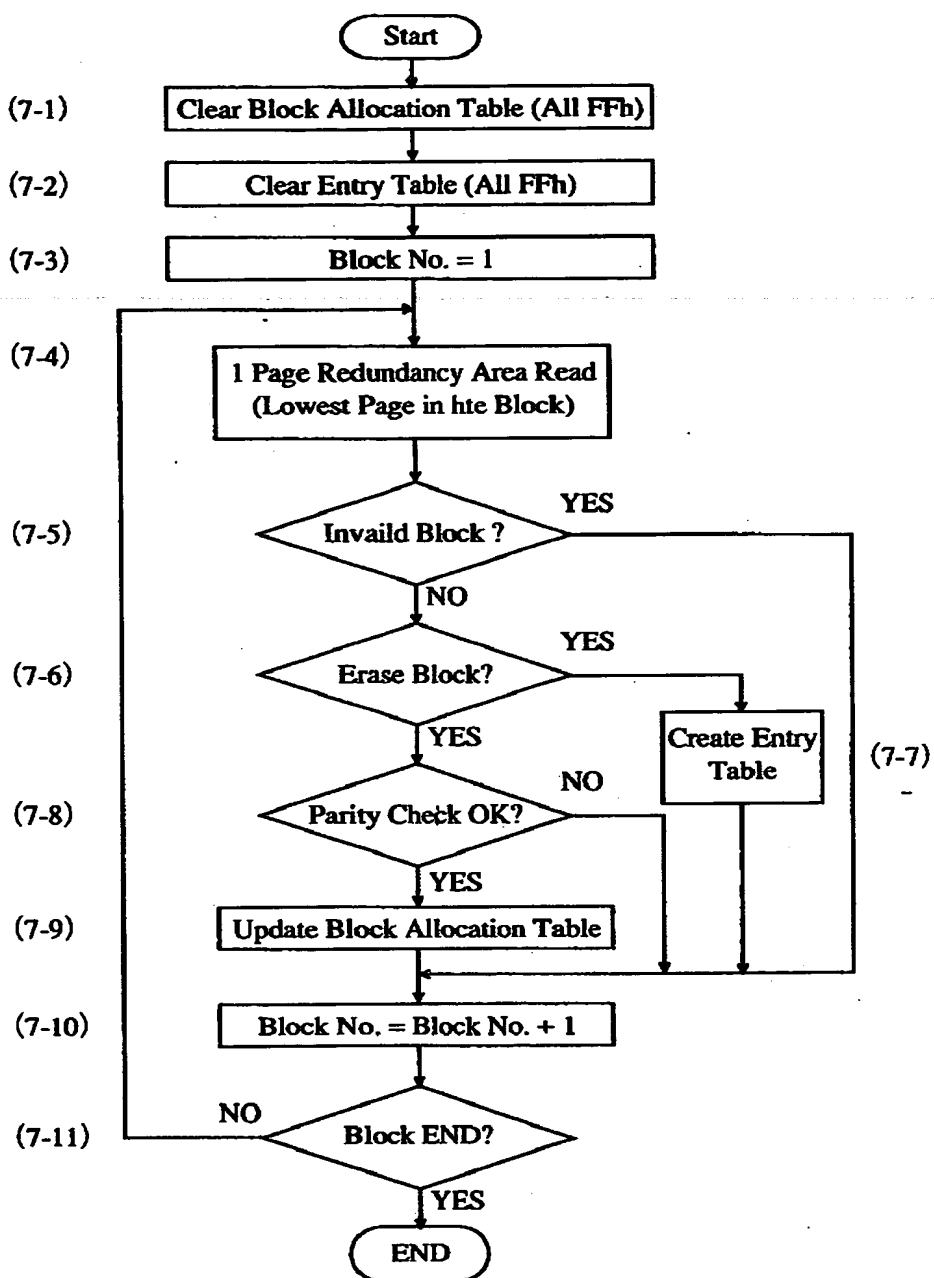
【図 5】

LP00=D (\*\*\*\*\*0、\*\*\*)、 LP01=D (\*\*\*\*\*1、\*\*\*)  
 LP02=D (\*\*\*\*\*0\*、\*\*\*)、 LP03=D (\*\*\*\*\*1\*、\*\*\*)  
 LP04=D (\*\*\*\*\*0\*\*、\*\*\*)、 LP05=D (\*\*\*\*\*1\*\*、\*\*\*)  
 LP06=D (\*\*\*\*0\*\*\*、\*\*\*)、 LP07=D (\*\*\*\*1\*\*\*、\*\*\*)  
 LP08=D (\*\*\*0\*\*\*\*、\*\*\*)、 LP09=D (\*\*\*1\*\*\*\*、\*\*\*)  
 LP10=D (\*\*0\*\*\*\*\*、\*\*\*)、 LP11=D (\*\*1\*\*\*\*\*、\*\*\*)  
 LP12=D (\*0\*\*\*\*\*、\*\*\*)、 LP13=D (\*1\*\*\*\*\*、\*\*\*)  
 LP14=D (0\*\*\*\*\*、\*\*\*)、 LP15=D (1\*\*\*\*\*、\*\*\*)  
 CP00=D (\*\*\*\*\*、\*\*0)、 CP01=D (\*\*\*\*\*、\*\*1)  
 CP02=D (\*\*\*\*\*、\*0\*)、 CP03=D (\*\*\*\*\*、\*1\*)  
 CO04=D (\*\*\*\*\*、0\*\*)、 CP05=D (\*\*\*\*\*、1\*\*)

【図6】



【図7】

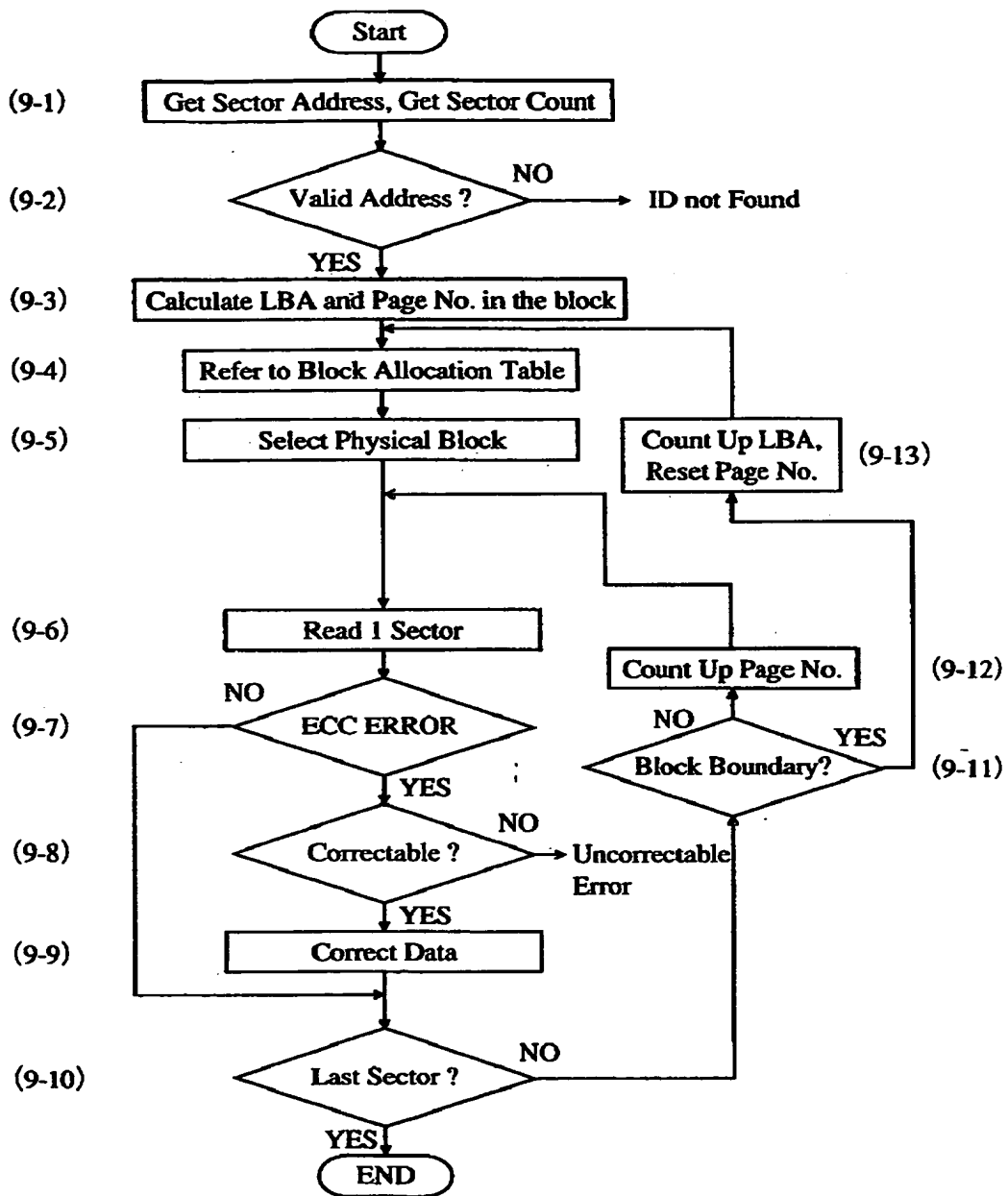


【図 8】

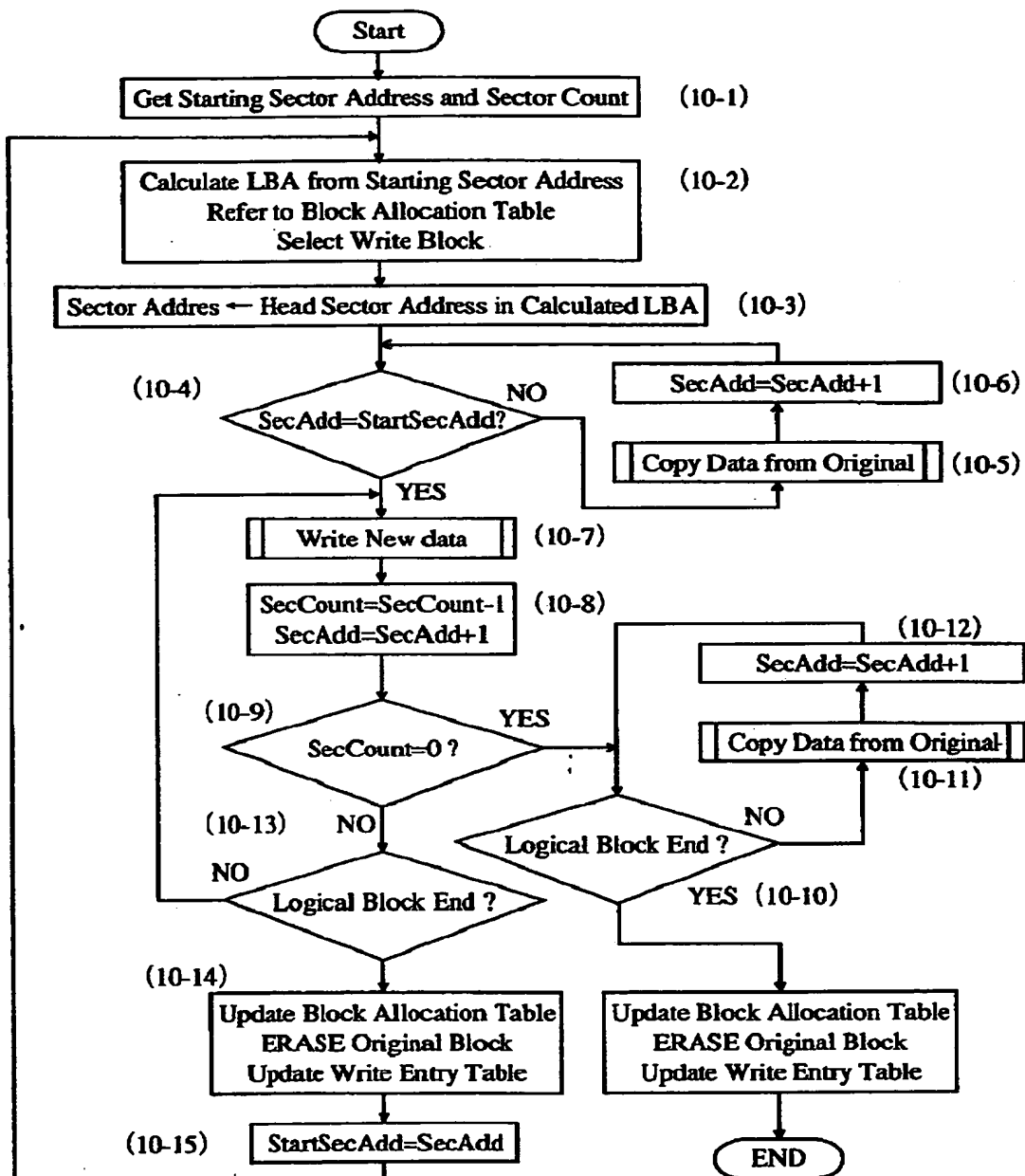
OFFSET		Byte
Byte0	(LBA=0)	Physical Block Area
Byte1	(LBA=1)	Physical Block Area
Byte2	(LBA=2)	Physical Block Area
⋮		
Byte498	(LBA=497)	Physical Block Area
Byte499	(LBA=498)	Physical Block Area
Byte500	(LBA=499)	Physical Block Area

1 Physical Block Area = 2 Physical Block

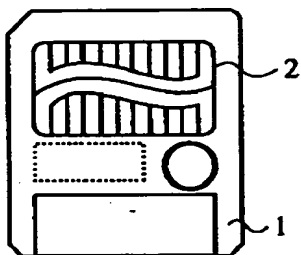
【図9】



【図10】



【図 1 1】



【図 1 2】

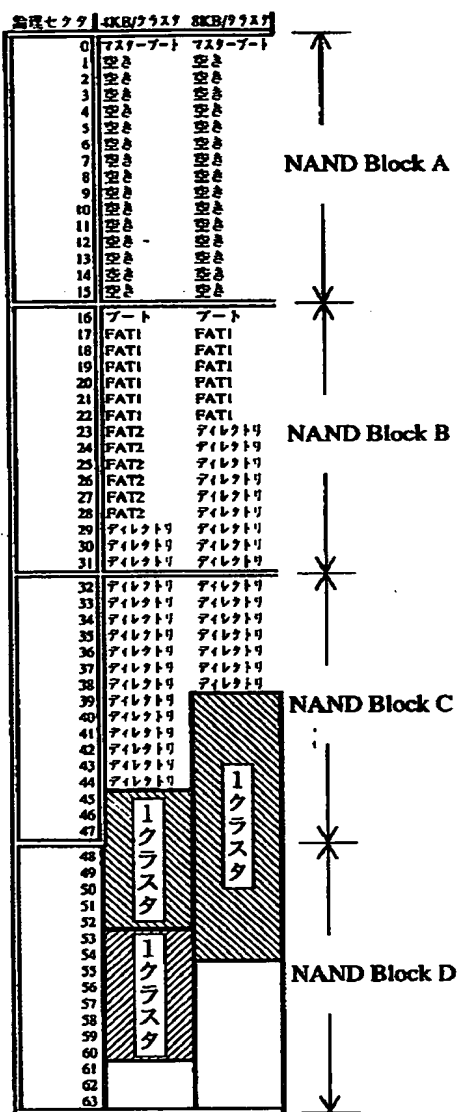
OFFSET	Upper Byte	Lower Byte
Word0 (LBA=0)	Physical Block Address上位	Physical Block Address下位
Word1 (LBA=1)	Physical Block Address上位	Physical Block Address下位
Word2 (LBA=2)	Physical Block Address上位	Physical Block Address下位
⋮	⋮	⋮
Word498 (LBA=497)	Physical Block Address上位	Physical Block Address下位
Word499 (LBA=498)	Physical Block Address上位	Physical Block Address下位
Word500 (LBA=499)	Physical Block Address上位	Physical Block Address下位

【図 1 3】

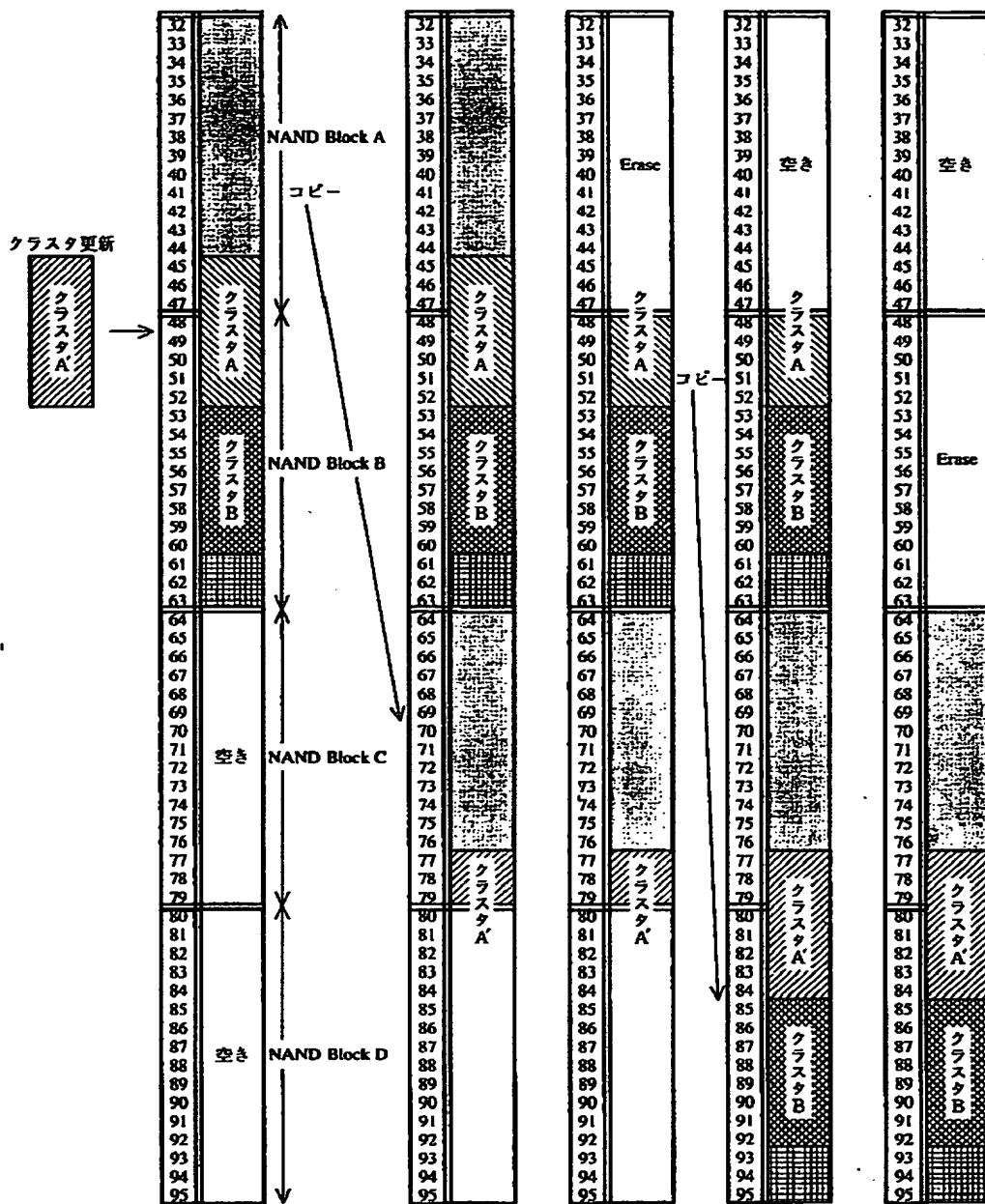
OFFSET	Upper Byte	Lower Byte
Word0 (LBA=0)	Physical Block Address上位	Physical Block Address下位
Word1 (LBA=1)	Physical Block Address上位	Physical Block Address下位
Word2 (LBA=2)	Physical Block Address上位	Physical Block Address下位
⋮	⋮	⋮
Word997 (LBA=997)	Physical Block Address上位	Physical Block Address下位
Word998 (LBA=998)	Physical Block Address上位	Physical Block Address下位
Word999 (LBA=999)	Physical Block Address上位	Physical Block Address下位



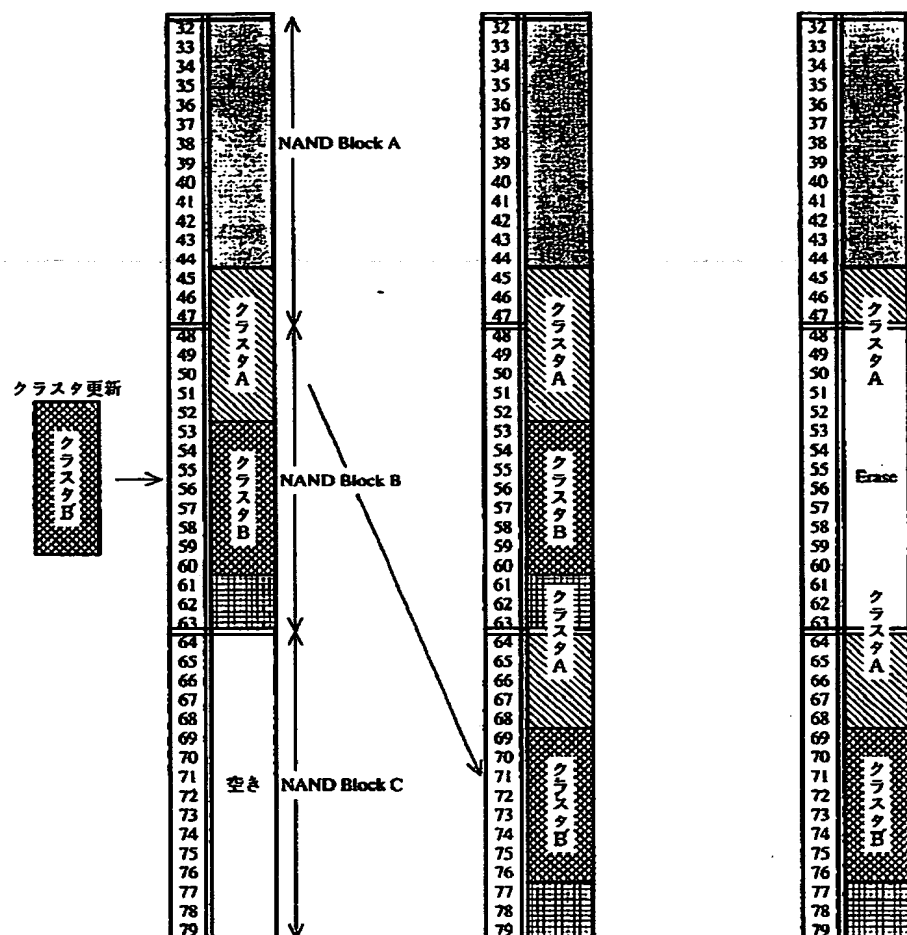
【図14】



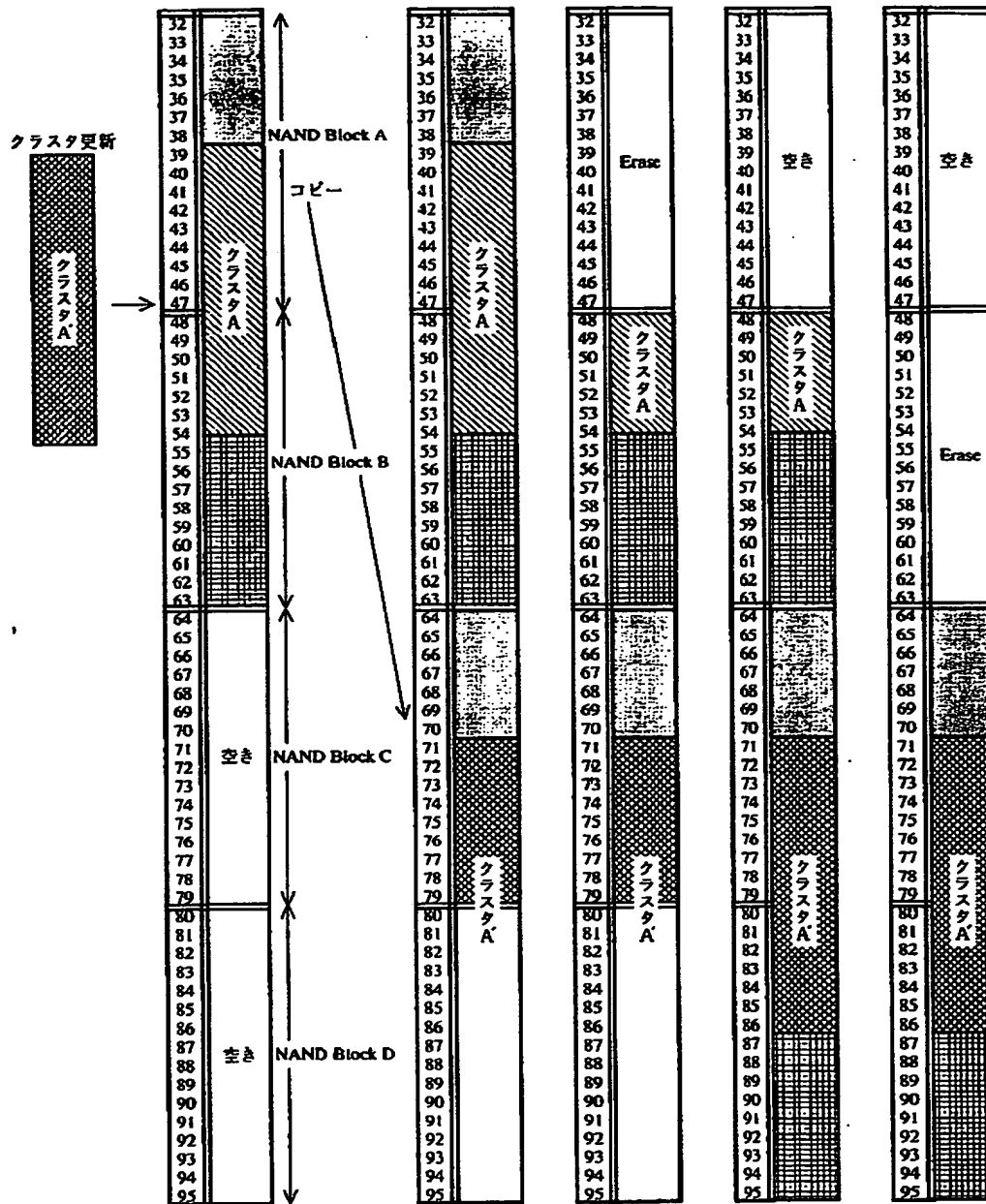
【图 15】



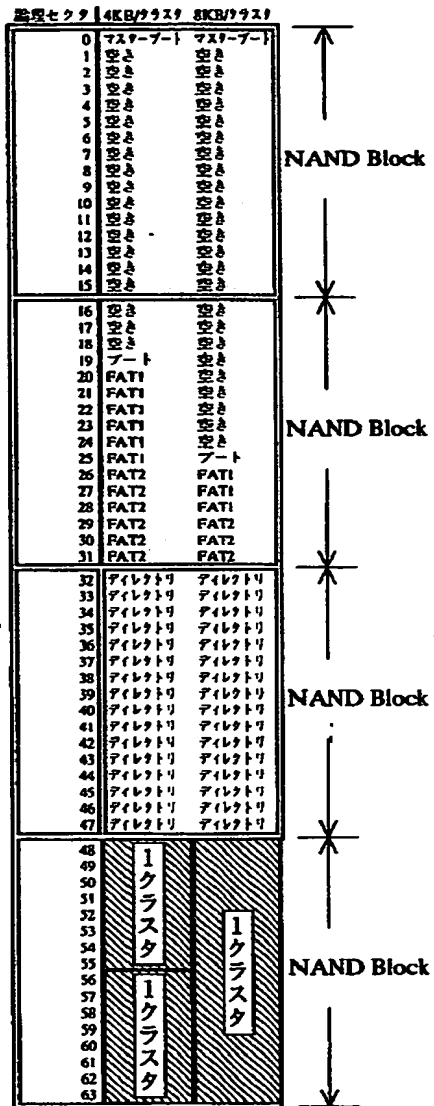
【図16】



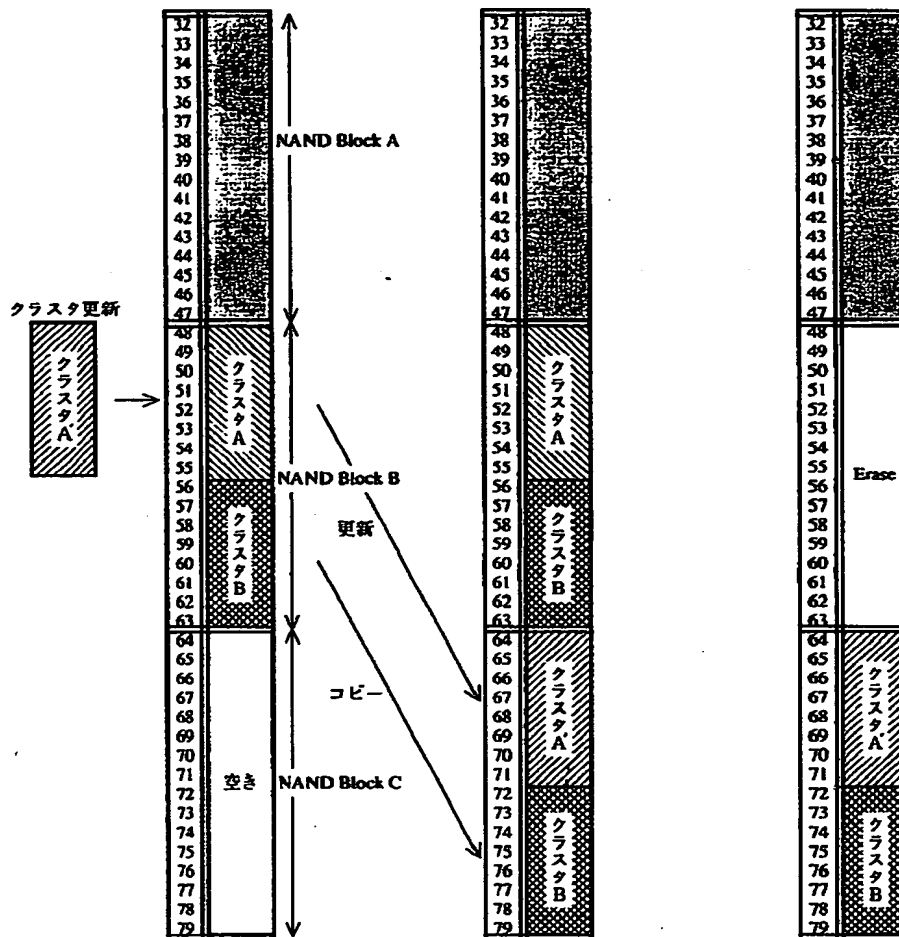
【図17】



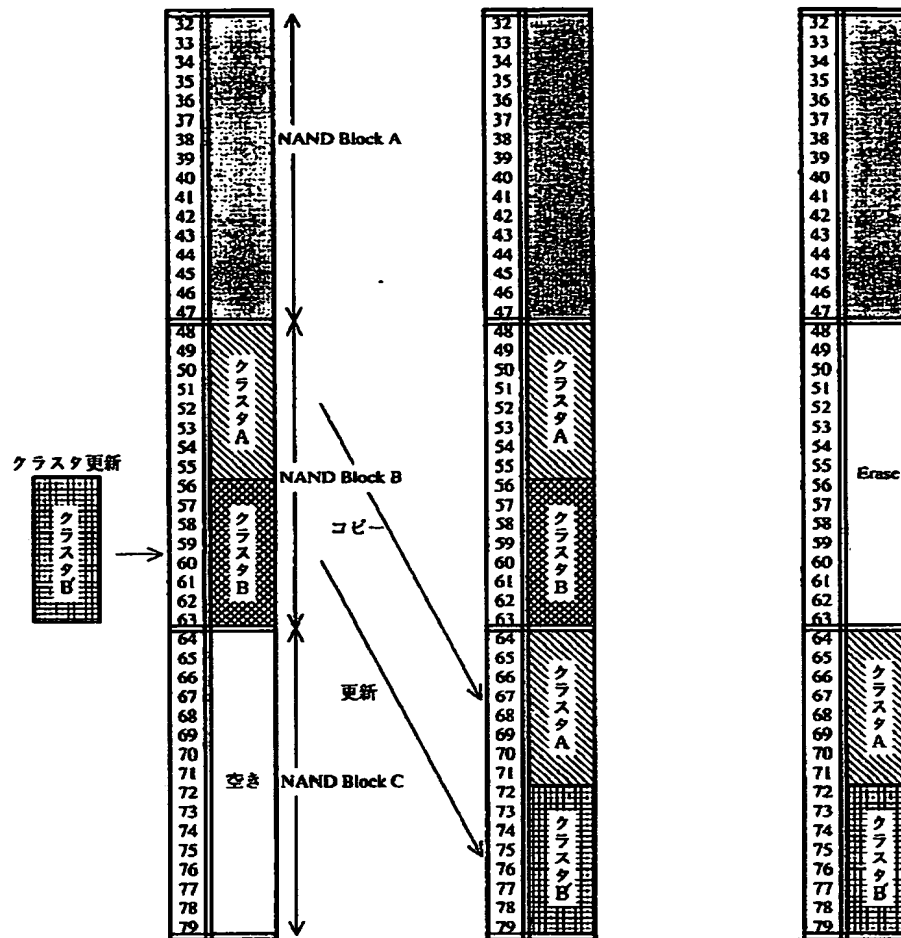
【図18】



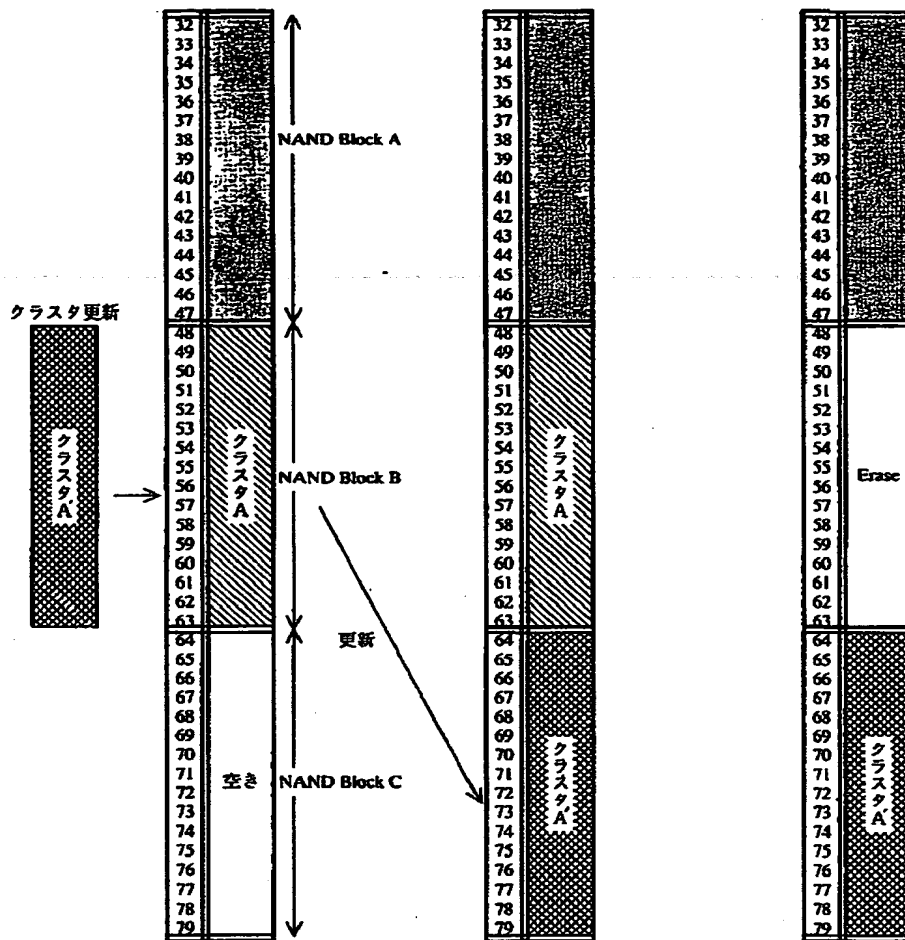
【図19】



【図20】



【図 21】





【図 2 2】

管理領域

File-1	
File-2	
File-3	
File-4	
...	
File-N	

File-1, File-4消去

管理領域

File-1	del. mark
File-2	
File-3	
File-4	del. Mark
...	
File-N	

データ領域

File-1
File-2
File-3
File-4

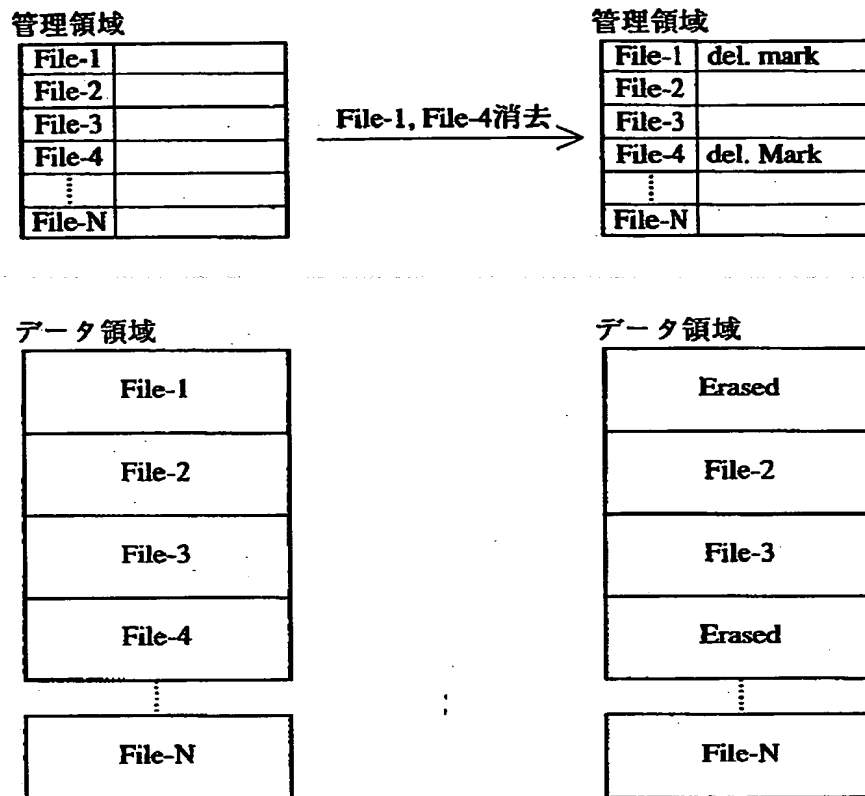
File-N
--------

データ領域

File-1
File-2
File-3
File-4

File-N
--------

【図23】



**【書類名】 要約書****【要約】**

**【課題】** セルアレイが複数の物理ブロックに分割され、各々の物理ブロックにシステムが管理する論理ブロックとの対応付けの情報を記憶メモリシステムにおいて、論理ブロックと物理ブロックとの対応関係を管理するためのテーブルに要するRAM領域の削減を目的とする。また、データ更新および消去後の書込みの高速化を目的とする。

**【解決手段】** 論理ブロック／物理ブロック変換テーブルにおいて、1個の論理ブロックに対して、複数の物理ブロックを含んだエリアを割り振り、メモリアクセス時には、そのエリア中の複数のブロックの記憶情報をサーチし、該エリア中から真の対応ブロックを選択することを特徴とするメモリシステムの制御方法を提供する。また、物理ブロック上でクラスタの区切れがブロックの区切れをまたがない制御方法を提供する。また、消去時に管理領域を開放すると同時にデータ領域の消去を行う制御方法を提供する。

**【選択図】** 図8

【書類名】 職権訂正データ  
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000003078

【住所又は居所】 神奈川県川崎市幸区堀川町72番地

【氏名又は名称】 株式会社東芝

【代理人】 申請人

【識別番号】 100083161

【住所又は居所】 東京都港区芝浦1丁目1番1号 株式会社東芝本社  
事務所内

【氏名又は名称】 外川 英明

出 願 人 履 歴 情 報

識別番号 [000003078]

1. 変更年月日 1990年 8月22日

[変更理由] 新規登録

住 所 神奈川県川崎市幸区堀川町72番地  
氏 名 株式会社東芝